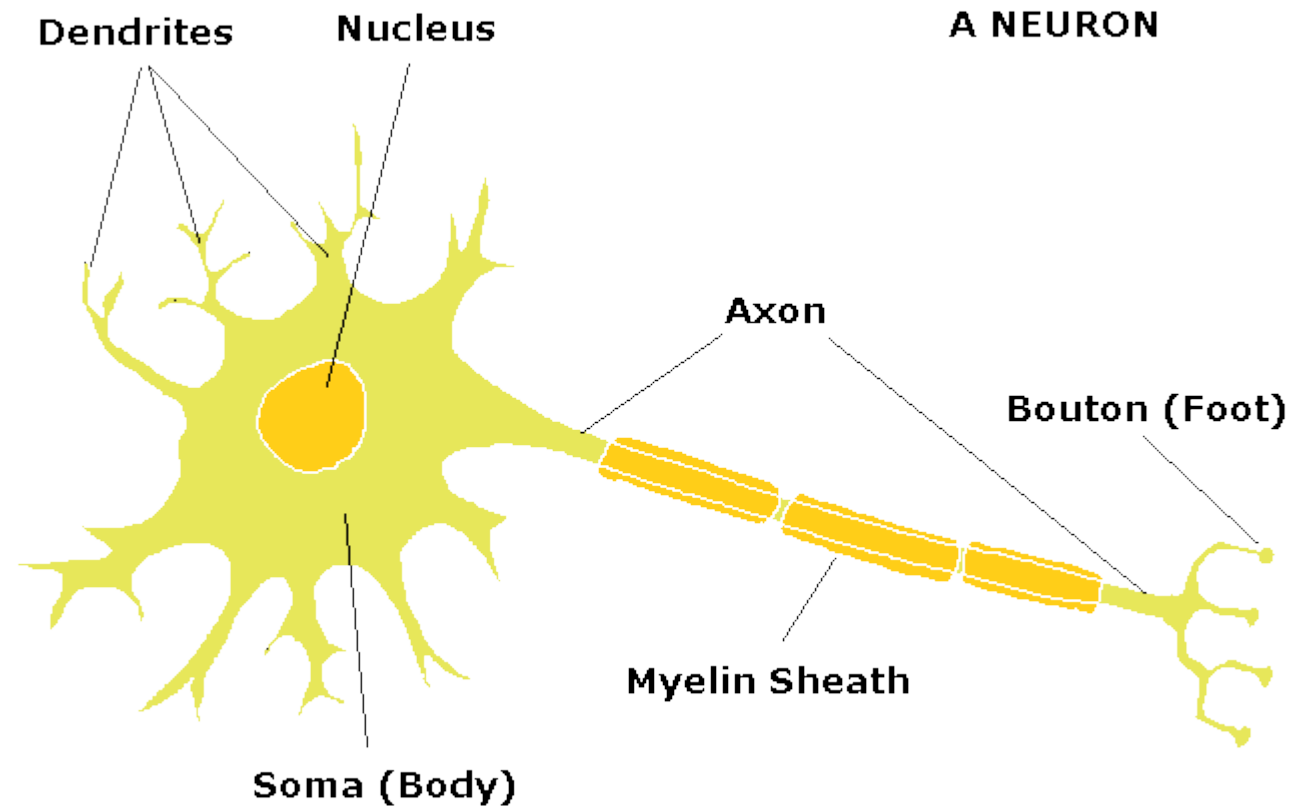
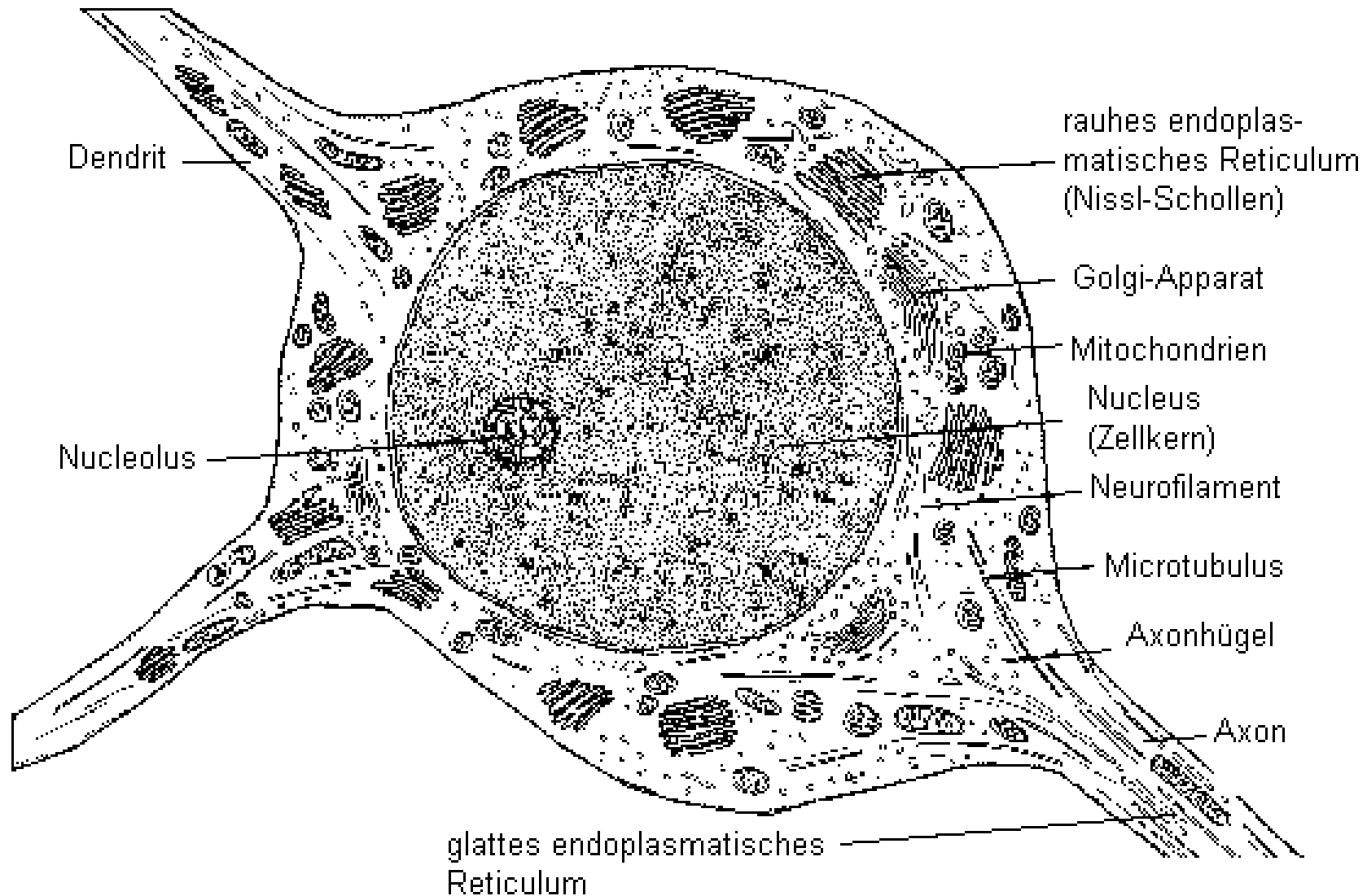


Lernen und Neuronale Netze



Biologisches Vorbild



Neuronale Netze

Eigenschaften biologischer Systeme, die man gerne übernehmen möchte:

- Fehlertoleranz
- Generalisierungsfähigkeit
- Lernen
- Selbstorganisation

Typen von Neuronalen Netzen

- supervised
- selbst-organisierend
- reinforcement

	Deutsch	Englisch	Beschreibung
Teacher	Überwachtes Lernen	Supervised	Es gibt einen desired Output, welchen das Netz lernen soll.
Reward	Bestärkendes Lernen	Reinforcement Learning	Es gibt nur eine Belohnung, die sagt, ob die Aktion gut oder schlecht war.
-	Unüberwachtes Lernen	Unsupervised	Es gibt gar keinen Output mehr. Wir können die Muster nur zu Clustern zusammenfassen

Lernen

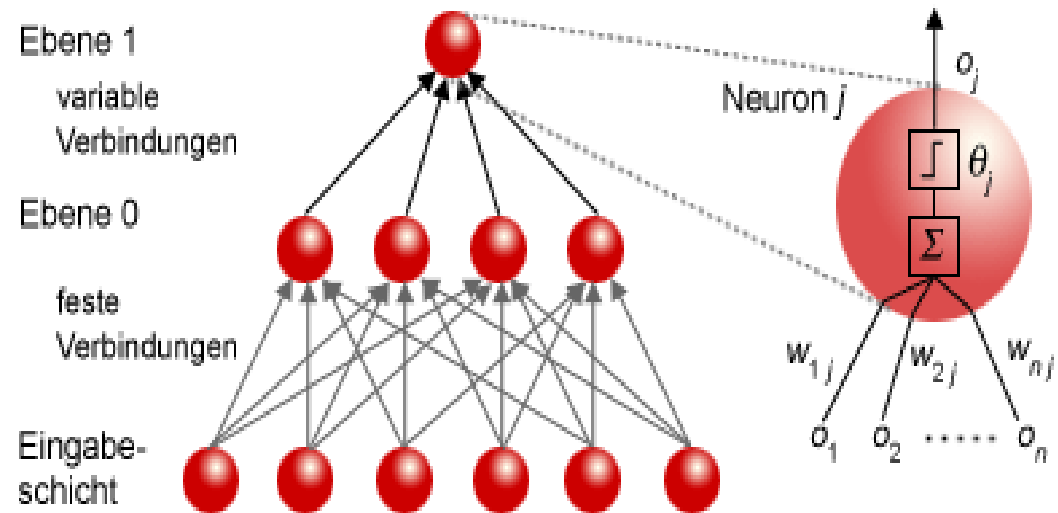
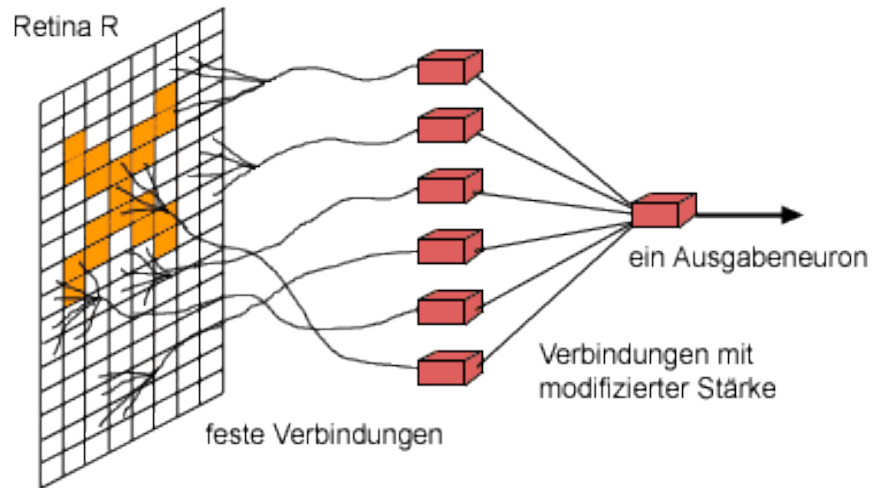
Hebbsche Lernregel (D. Hebb, 1949)

"Wenn ein Axon der Zelle A nahe genug ist, um eine Zelle B zu erregen und wiederholt oder dauerhaft sich am Feuern beteiligt, geschieht ein Wachstumsprozeß oder metabolische Änderung in einer oder beiden Zellen dergestalt, daß A's Effizienz als eine der auf B ... feuernden Zellen anwächst."

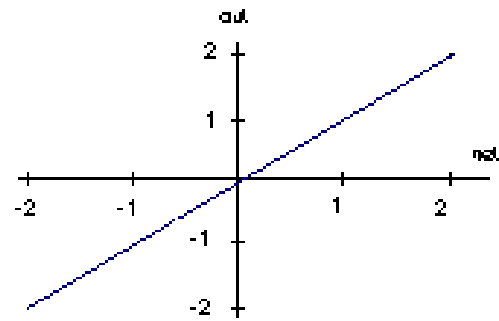
$$\Delta w_{ij} = \eta \cdot o_i \cdot a_j \text{ und } w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}$$



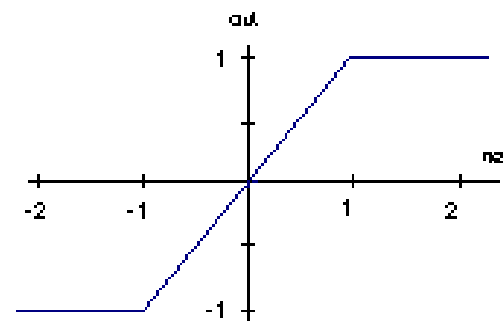
Perzeptron



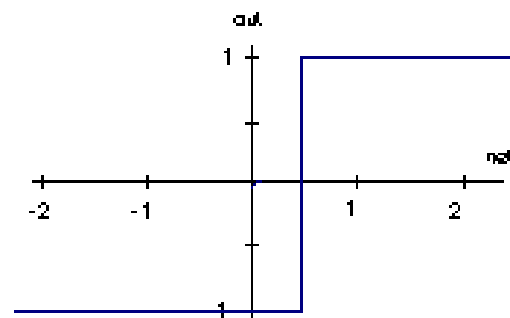
Aktivierungsfunktionen



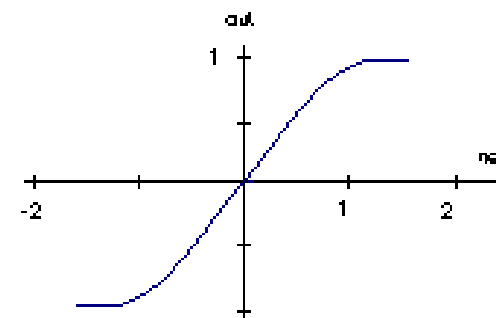
Identität ($o_j = net_j$)



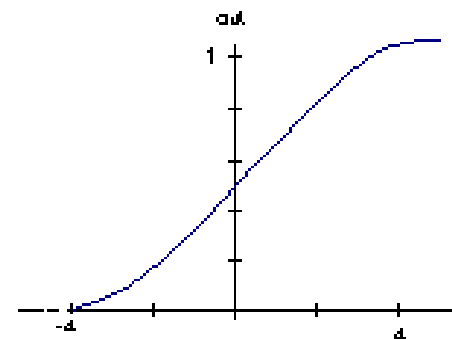
linear bis Sättigung



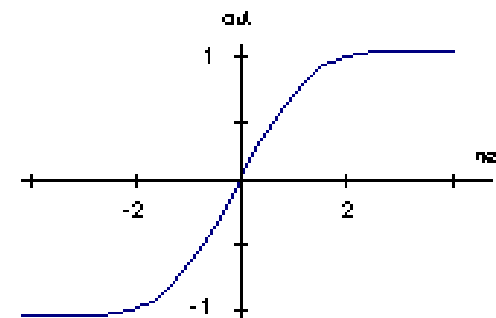
binäre Schwellenwertfkt.



sin(x) bis Sättigung

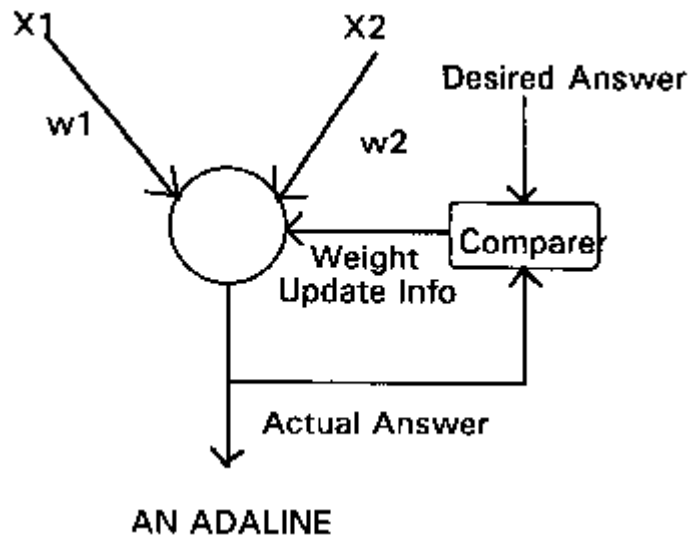


$(1 / 1 + \exp(-x))$



$\tanh(x)$

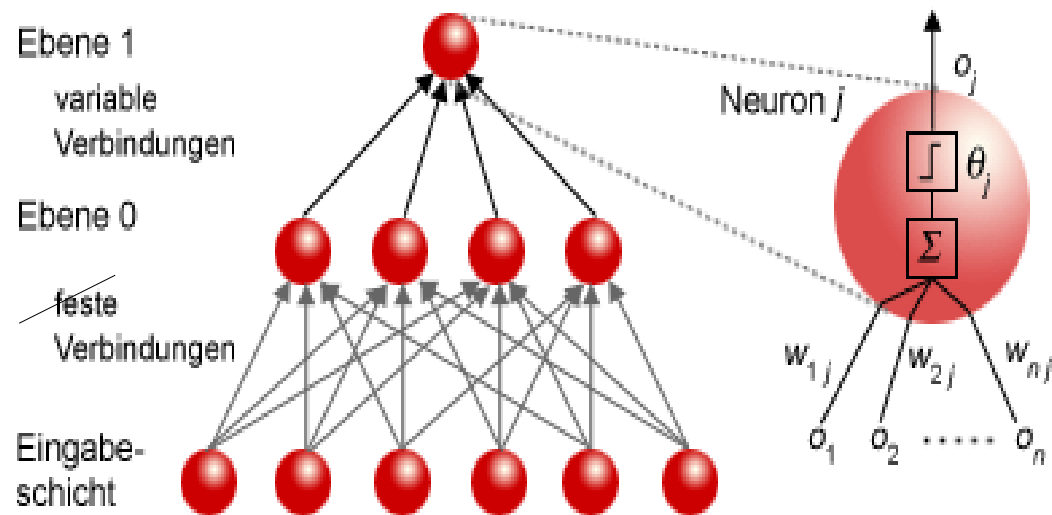
Delta Regel



$$\Delta w_{ik} = \eta (d_k - o_k) \tilde{o}_i$$

MLP mit Backpropagation

Multi-Layer-Perceptron mit Input-, Hidden- und Output Schicht



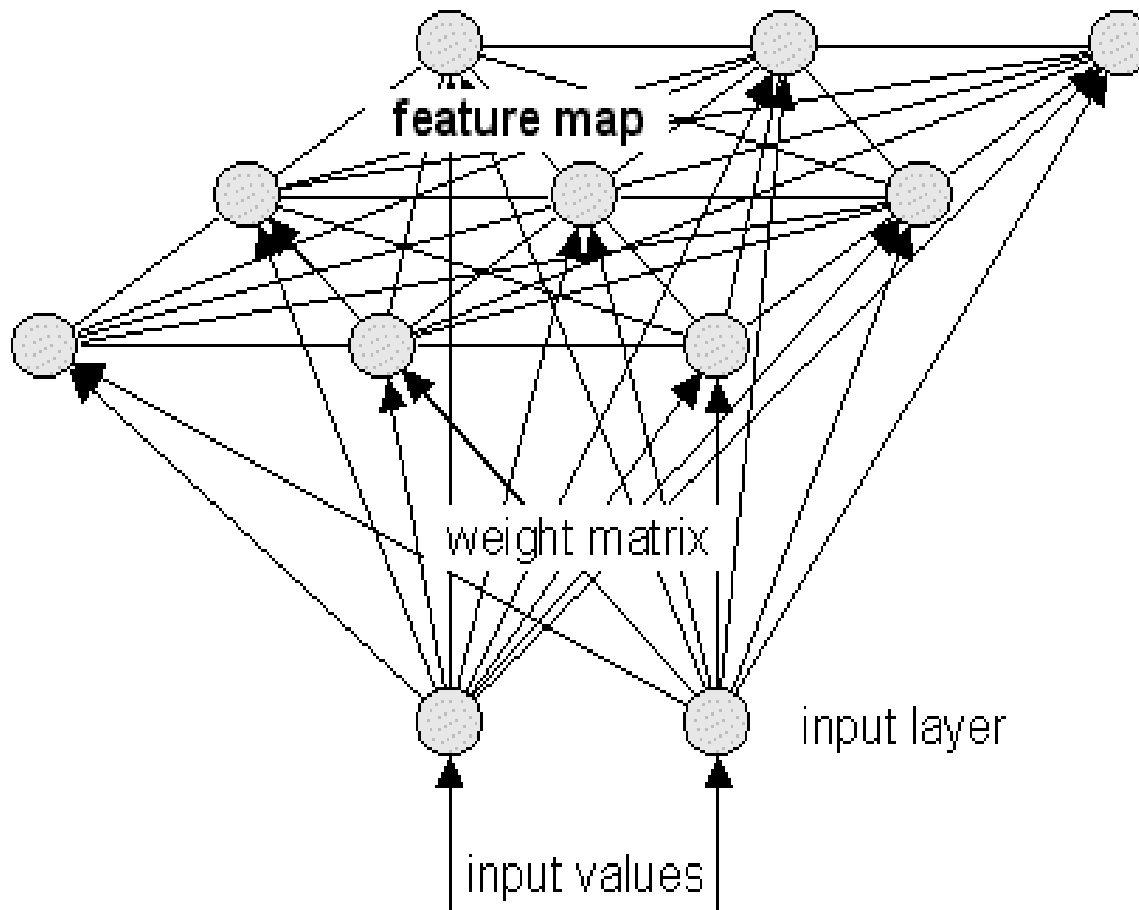
MLP mit Backpropagation

wichtig bei MLPs:

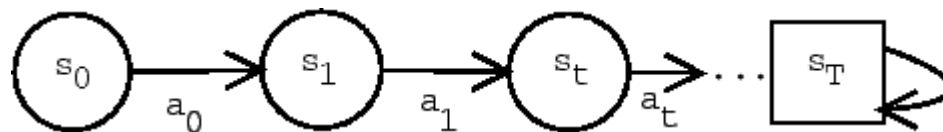
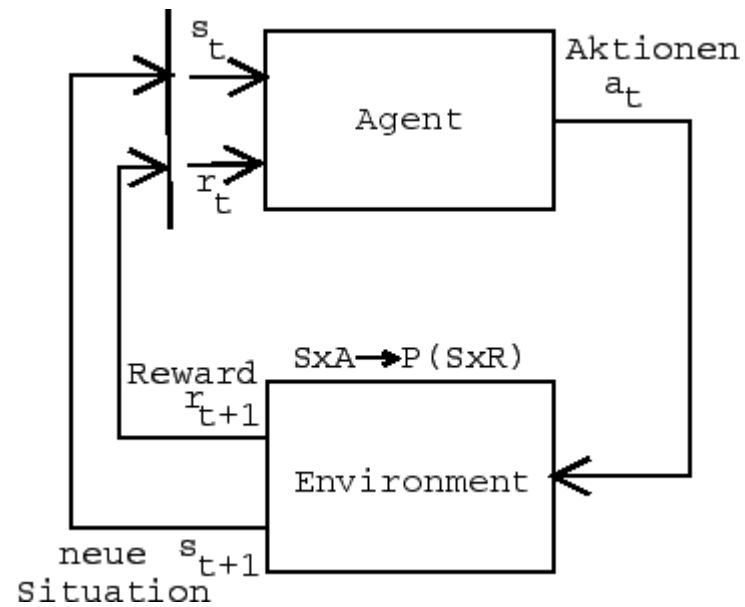
- Wahl der Anzahl der hidden layers (meist eine)
- Wahl der Anzahl der hidden neurons (kommt auf Problem und Anzahl der I/O Neurons an)
- Kodierung des gewünschten outputs (z.B. population coding)
- Kodierung des gewünschten Inputs (z.B. image features statt pixel)

Unsupervised Networks

Beispiel: Kohonen Self-Organising Maps



Reinforcement Lernen



Reinforcement Lernen

Der **Polebalancer** besteht aus einem Wagen, der nach links und rechts per Bang-Bang-Control beschleunigt werden kann. Dabei muss er immer in eine Richtung beschleunigt werden. Er kann nicht einfach stehen gelassen werden. Der Wagen hat eine Position auf der Schiene mit einer Geschwindigkeit. An den Seiten der Schiene, auf der der Wagen steht, befinden sich Begrenzungen. Auf dem Wagen befindet sich ein Pole an einem Gelenk. Dieser hat einen Winkel zur Senkrechten und eine Geschwindigkeit. Ziel des Pole Balancers ist es, den Stab gerade zu halten. Wir können dies mit Reinforcementlearning mit einem pure negativ reward tun. Wir brauchen, um den Pole-Balancer zu steuern exakt ein Neuron. Um ihn zu trainieren, ein weiteres.

